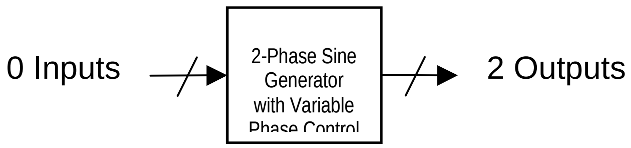
2-Phase Sine Generator with Variable Phase Control

DSP-based

RIDE	Block Diagram	ImageDSP	VAB for C2000	VAB for C5000	VAB for C6000	VAB for UNIV
●			●			

Component Name: 2-Phase Sine Generator with Variable Phase Control



Library	Digital Motor Control Library	
Group	Signal Generators	
File Name	SINCOSPH.DLL	Revision: 4

Overview: Generates two sine waves with variable magnitude, frequency, and phase difference

Outputs: The first output channel from this block component is the first sine waveform. The second output channel is the second sine waveform. The phase difference between first and second waveform will be equal to the specified Phase Shift angle. The format for both outputs is Q15 with a range between 8000h-7FFFh.

Parameters:
The following parameters of '2-Phase Sine Generator with Variable Phase Control' may be modified:

Gain-magnitude The Gain-magnitude parameter is used to specify the magnitude of the two sine waveforms. The format for this parameter is Q15 and in the range of 0-7FFFh.

Frequency The Frequency parameter is used to specify the frequency of the two sine waveforms. The format for this parameter is Q15 and in the range of 0-7FFFh.

Phase Shift The Phase Shift parameter is used to specify the phase difference between the two sine waveforms. The format for this parameter is Q0 and in the range of 0-180.

Precision This block component requires that the precision field be set for integer precision. This parameter is not user-selectable.

> *Note - All parameters except the 'Precision' parameter of 2-Phase Sine Generator with Variable Phase Control are able to be set by either the user, or by the output of any other DSP-based component.*

Description:
This block component generates two sine waves with variable magnitude (gain_cs), frequency (freq), and phase difference (phase). The two sine waves are output on channels 1 (sine_a1) and 2 (sine_a2). The maximum magnitude of these waves set by the Gain-magnitude user-parameter. The frequency of the waves is set by the Frequency user-parameter, and the phase difference is set by the Phase Shift user-parameter.

The generation of the sine wave is performed using a look up table. To be able to control the frequency of sine waves, a method based on the modulo mathematical operation is used. A 16 bit software counter is used to determine the location of the next value of the sine waves. A step value is added to the counter everytime a new value from the sine table is to be loaded. By changing the value of the step, one can accurately control the frequency of the sine wave.

Although a 16 bit counter is used, the upper byte determines the location of the next sine value to be used; thus, by changing how quickly values overflow from the lower byte (i.e. manipulating the step value), the frequency of the sine wave can be changed. The modulo mathematical operation is used when there is overflow in the accumulator from the lower word to the upper word. When an overflow occurs, only the remainder (lower word) is stored.

For example, the counter is set to 0000h and the step value is set to 40h. Every time a value is to be looked up in the table, the value 40h is added to the counter; however, since the upper byte is used as the pointer on the look up table, the first, second, and third values will point to the same location. In the fourth step, which results in an overflow into the upper byte, the value that is loaded will change. Since the upper byte is used as the pointer, the lookup table has 256 values, which is equivalent to the number of possibilities for an 8-bit number (0-255). Additionally, since the upper word of the accumulator is disregarded, the pointer for the sine lookup table does not need to be reset.

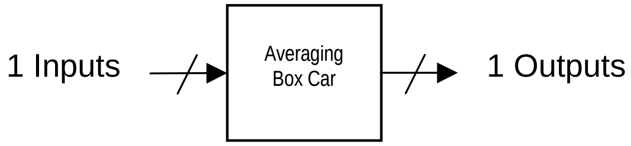
The step size controls the frequency that is output; as a result, the larger the step, the quicker the overflow into the upper byte, and the faster the pointer traverses through the sine lookup table.

Averaging Box Car

DSP-based

RIDE	Block Diagram	ImageDSP	VAB for C2000	VAB for C5000	VAB for C6000	VAB for UNIV
●			●			

Component Name: Averaging Box Car



Library	Digital Motor Control Library	
Group	Arithmetic Functions	
File Name	BC_CALC.DLL	Revision: 4.00

Overview: Calculates the boxcar average of an input signal

Inputs: The input channel to this block component is expected to be a Q15 data value. The framesize for the input must be equal to a length of 1.

Outputs: The output channel from this block component is the computed boxcar averaged signal. The framesize for the output is of length 1, and the data format is Q15.

Parameters:
The following parameters of 'Averaging Box Car' may be modified:

Box Car Scaler (Q15)	This parameter is used to specify the scaler value applied to the boxcar average. The format of this parameter is expected to be Q15.
Precision	This block component requires that the precision field be set for integer precision. This parameter is not user-selectable.

Note - All parameters except the 'Precision' parameter of Averaging Box Car are able to be set by either the user, or by the output of any other DSP-based component.

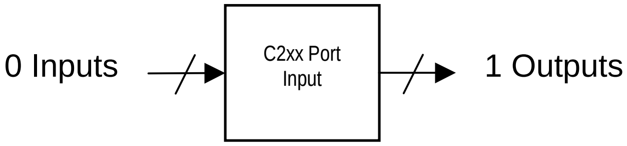
Description:
This block component calculates the average value of an input sample. The output can be rescaled. The size of buffer used for storing the averaging data is set for 64 samples.

C2xx Port Input

DSP-based

RIDE	Block Diagram	ImageDSP	VAB for C2000	VAB for C5000	VAB for C6000	VAB for UNIV
●					●	

Component Name: C2xx Port Input



Library	Native Library	
Group	C2xx Hardware Functions	
File Name	RTINPT.DLL	Revision: 4.00

Overview: Input data from specified port

Parameters:
The following parameters of 'C2xx Port Input' may be modified:

Port Address	The port from which to read. May be specified in hexadecimal or decimal.

Note - All parameters except the 'Precision' parameter of C2xx Port Input are able to be set by either the user, or by the output of any other DSP-based component.

Description:
Reads from the specified port address and outputs the result to the output channel. Since this is a native block, only one sample is created each time the block is executed.

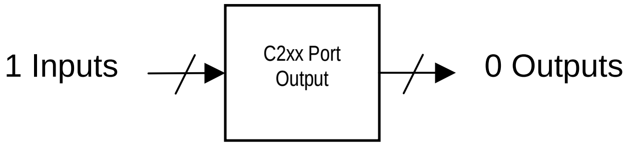
See Also: C2xx Port Output

C2xx Port Output

DSP-based

RIDE	Block Diagram	ImageDSP	VAB for C2000	VAB for C5000	VAB for C6000	VAB for UNIV
●					●	

Component Name: C2xx Port Output



Library	Native Library
Group	C2xx Hardware Functions
File Name	RTOUTPT.DLL Revision: 4.00

Overview: Output data to specified port

Inputs: The value to be output on the specified port. The port address is specified in the parameters for the block.

Parameters:
The following parameters of 'C2xx Port Output' may be modified:

Port Address	The output port address to which the data will be written. Typically 0xC (the location of the LEDs).

Note - All parameters except the 'Precision' parameter of C2xx Port Output are able to be set by either the user, or by the output of any other DSP-based component.

Description:
Ouputs the data on the input channel to the specified port. For example, to control the LEDs on the board, specify port 0xC (the address of the LEDs). Since this block is a native block, only the first value on the input channel is used.

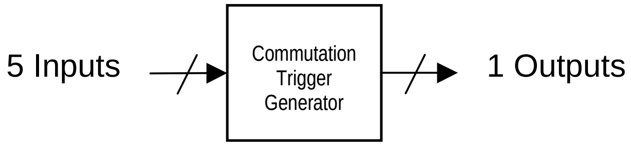
See Also: C2xx Port Output

Commutation Trigger Generator

DSP-based

RIDE	Block Diagram	ImageDSP	VAB for C2000	VAB for C5000	VAB for C6000	VAB for UNIV
•			•			

Component Name: Commutation Trigger Generator



Library	Digital Motor Control Library	
Group	AC Induction Motor Functions	
File Name	COMTN_TRIG.DLL	Revision: 4.00

Overview: Determines the Bemf zero crossing points of a 3-ph BLDC motor

Inputs: The first input channel to this block component is the commutation state pointer input. This is used for Bemf zero-crossing point calculation for the appropriate motor phase. The format for this input is expected to be Q0 with a range of 0-5. The second through fourth input channels are the Motor phase voltages referenced to GND. These inputs are expected to be Q15 with a range of 0-7FFF. The fifth input channel is a virtual timer that is used for commutation delay angle calculation. The format of this input is expected to be Q0 with a range of 0-7FFF.

Outputs: The output channel for this block component is the Commutation trigger output. The format for this output is Q0 with a range of 0-7FFF.

Parameters:
The following parameters of 'Commutation Trigger Generator' may be modified:

Precision This block component requires that the precision field be set for integer precision. This parameter is not user-selectable.

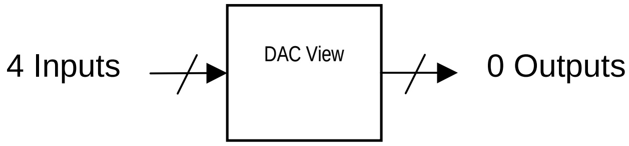
Note - All parameters except the 'Precision' parameter of Commutation Trigger Generator are able to be set by either the user, or by the output of any other DSP-based component.

Description:
This block component determines the Bemf zero crossing points of a 3-ph BLDC motor based on motor phase voltage measurements and then generates the commutation trigger points for the 3-ph power inverter switches.

DAC View

						DSP-based
RIDE	Block Diagram	ImageDSP	VAB for C2000	VAB for C5000	VAB for C6000	VAB for UNIV
•			•			

Component Name: DAC View



Library	Digital Motor Control Library	
Group	AC Induction Motor Functions	
File Name	DACVIEW.DLL	Revision: 4.00

Overview: Generates D/A output

Inputs: The four input channels to this block component represent the Q15 data to undergo D/A conversion. These input channels will be applied to the 12-bit DAC of the 24x/24xx EVM.

Parameters:
The following parameters of 'DAC View' may be modified:

Scale	The Scale parameter controls the Q12 scale value that is applied to the input channels. The default value used is 2000 which correponds to value of approximately 0.5.
Precision	This block component requires that the precision field be set for integer precision. This parameter is not user-selectable.

Note - All parameters except the 'Precision' parameter of DAC View are able to be set by either the user, or by the output of any other DSP-based component.

Description:
This block component will convert 4 input channels of Q15 data into its equivalent Q0 format that spans the full input range of a 12-bit DAC. The output channels are then applied to the 12-bit DAC on the 24x/24xx EVM.

Note for 'DAC View':
Since this is a Native function it has been optimized for single sample processing and is quite efficient, especially for control applications and systems which require small amounts of processing delay.

Although it will interface with other non-Native functions (i.e., Vector functions), the inputs of this component are treated as scalar values and not as vectors, so this function will only process the first sample of a vector fed into it.

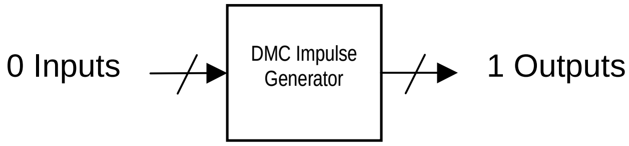
See Also: Absolute Memory Read; Absolute Memory Write; Native EvmF240 ADC; Native EvmF240 DAC
Example Worksheets: RampDAC

DMC Impulse Generator

DSP-based

RIDE	Block Diagram	ImageDSP	VAB for C2000	VAB for C5000	VAB for C6000	VAB for UNIV
●			●			

Component Name: DMC Impulse Generator



Library	Digital Motor Control Library	
Group	Signal Generators	
File Name	IMPULSE2.DLL	Revision: 4

Overview: Generates a periodic impulse

Outputs: The output channel from this block component is the Impulse generator output. The format of this data is Q0 with either a value of 0, or a value of 7FFF.

Parameters:
The following parameters of 'DMC Impulse Generator' may be modified:

Period The Period parmater is used to specify the period of output impulses in number of sampling cycles. The format for this parameter is Q0 in a range of 0-7FFF.

Precision This block component requires that the precision field be set for integer precision. This parameter is not user-selectable.

Note - All parameters except the 'Precision' parameter of DMC Impulse Generator are able to be set by either the user, or by the output of any other DSP-based component.

Description:
This block component implements a periodic impulse function. The output channel (ig_out) is set to 7FFF for 1 sampling period. The period of the output signal ig_out is specified by the Period user-parameter.

This block component will implement the following equation:

ig_out = 7FFF, for t = n . Tout, n = 1, 2, 3, ...
= 0, otherwise

where,

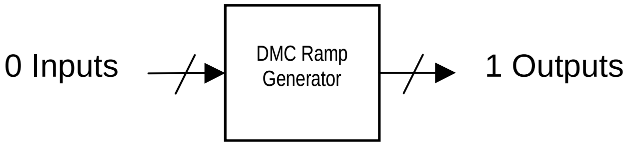
Tout = Time period of output pulses = Period x Ts
Ts = Sampling time period

DMC Ramp Generator

DSP-based

RIDE	Block Diagram	ImageDSP	VAB for C2000	VAB for C5000	VAB for C6000	VAB for UNIV
●			●			

Component Name: DMC Ramp Generator



Library	Digital Motor Control Library	
Group	AC Induction Motor Functions	
File Name	DMC RAMP.DLL	Revision: 4.00

Overview: Generates a ramp with the given frequency, gain and offset.

Outputs: The output channel of this block component is a Normalized Ramp. The format of the output is Q15 with a range of 0-7FFF.

Parameters:
The following parameters of 'DMC Ramp Generator' may be modified:

freq	The Freq parameter will specify the Normalized frequency of the ramp signal. The format is expected to be in Q15 with a range of 0-7FFF.
freq_max	The Freq Max parameter is used to specify the maximum step value for initialization of the maximum ramp frequency. The default value is set to 1000 to generate a maximum frequency of 305.2Hz using a 20kHz sampling loop. The format of this parameter is expected to be Q0.
gain	The Gain parameter is used to select the Normalized slope of the generated ramp signal. The format of this parameter is expected to be Q15 with a range of 0-7FFF.
offset	The Offset parameter is used to specify the Normalized DC offset in the genrated ramp signal. The format of this parameter is expected to be Q15 with a range of 0-7FFF.

Precision	This block component requires that the precision field be set for integer precision. This parameter is not user-selectable.

Note - All parameters except the 'Precision' parameter of DMC Ramp Generator are able to be set by either the user, or by the output of any other DSP-based component.

Description:
This block component generates a normalized ramp output of adjustable gain, frequency and dc offset.

In this implementation the frequency of the ramp output is controlled by a precision frequency generation algorithm which relies on the modulo nature (i.e. wrap-around) of finite length variables in 24x/24xx. One such variable, called alpha_rg (a data memory location in 24x/24xx) in this implementation, is used as a modulo-16 variable to control the time period (1/frequency) of the ramp signal. Adding a fixed step value (step_angle_rg) to this variable causes the value in alpha_rg to cycle at a constant rate through 0 to FFFFh. At the end limit the value in alpha_rg simply wraps around and continues at the next modulo value given by the step size. The rate of cycling through 0 to FFFFh is very easily and accurately controlled by the value of the step size.

Note for 'DMC Ramp Generator':
Since this is a Native function it has been optimized for single sample processing and is quite efficient, especially for control applications and systems which require small amounts of processing delay.

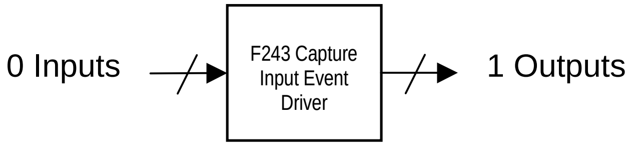
Although it will interface with other non-Native functions (i.e., Vector functions), the inputs and outputs of this component are treated as scalar values and not as vectors, so this function will only process the first sample of a vector fed into it, and its output framesize is always inherently 1.

F243 Capture Input Event Driver

DSP-based

RIDE	Block Diagram	ImageDSP	VAB for C2000	VAB for C5000	VAB for C6000	VAB for UNIV
●			●			

Component Name: F243 Capture Input Event Driver



Library	Digital Motor Control Library	
Group	AC Induction Motor Functions	
File Name	F243_CE_DRV.DLL	Revision: 4.00

Overview: Provides instantaneous time-stamp value captured on the occurrence of an event.

Outputs: The output channel from this block component is an integer time-stamp value read from the timer assigned to the capture unit. The format is expected to be Q0 with a range of 8000-7FFF.

Parameters:
The following parameters of 'F243 Capture Input Event Driver' may be modified:

Precision	This block component requires that the precision field be set for integer precision. This parameter is not user-selectable.

*Note - All parameters except the 'Precision' parameter of F243 Capture Input Event Driver are able to be set by either the user, or by the output of any other **DSP-based** component.*

Description:
This block component provides the instantaneous value of the selected time base (GP Timer) that is captured on the occurrence of an event. Such events can be any specified transition of a signal applied at the event manager (EV) capture input pins of 24x/24xx devices.

Note for 'F243 Capture Input Event Driver':
Since this is a Native function it has been optimized for single sample processing and is quite efficient, especially for control applications and systems which require small amounts of processing delay.

Although it will interface with other non-Native functions (i.e., Vector functions), the inputs and outputs of this component are treated as scalar values and not as vectors, so this function will only process the

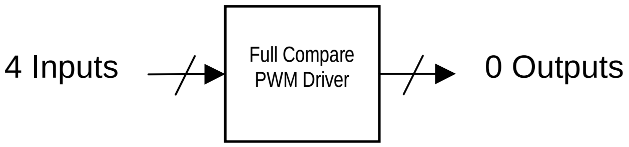
first sample of a vector fed into it, and its output framesize is always inherently 1.

Full Compare PWM Driver

DSP-based

RIDE	Block Diagram	ImageDSP	VAB for C2000	VAB for C5000	VAB for C6000	VAB for UNIV
●			●			

Component Name: Full Compare PWM Driver



Library	Digital Motor Control Library	
Group	AC Induction Motor Functions	
File Name	FC_PWM_DRV.DLL	Revision: 4.00

Overview: Calculates the compare values for generating PWM outputs using the full compare unit in 24x EVM

Inputs: The first three input channels are the duty ratios for full compare units 1,2, & 3. The fourth input channel is the PWM period modulation function. All inputs are expected to be Q15 numbers (8000-7FFF). The block component will use these inputs to compute the full compare PWM outputs from the 24x/24xx device.

Parameters:
The following parameters of 'Full Compare PWM Driver' may be modified:

Maximum Period The Maximum Period parameter is used to adjust the PWM period setting. Modify this parameter for different PWM frequency. The format of this paremeter is expected to be Q0, and the range is application dependent.

Precision This block component requires that the precision field be set for integer precision. This parameter is not user-selectable.

Note - All parameters except the 'Precision' parameter of Full Compare PWM Driver are able to be set by either the user, or by the output of any other DSP-based component.

Description:
This block component uses the duty ratio information and calculates the compare values for generating PWM outputs. The compare values are used in the full compare unit in 24x/24xx event manager (EV). This also allows PWM period modulation.

Note for 'Full Compare PWM Driver':

Since this is a Native function it has been optimized for single sample processing and is quite efficient, especially for control applications and systems which require small amounts of processing delay.

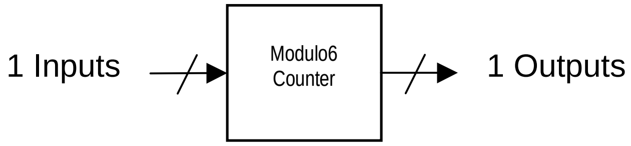
Although it will interface with other non-Native functions (i.e., Vector functions), the inputs and outputs of this component are treated as scalar values and not as vectors, so this function will only process the first sample of a vector fed into it, and its output framesize is always inherently 1.

Modulo6 Counter

DSP-based

RIDE	Block Diagram	ImageDSP	VAB for C2000	VAB for C5000	VAB for C6000	VAB for UNIV
•			•			

Component Name: Modulo6 Counter



Library	Digital Motor Control Library	
Group	Arithmetic Functions	
File Name	MOD6_CNT.DLL	Revision: 4.00

Overview: Implements a modulo 6 counter

Inputs: The input channel to this block component is the Modulo 6 counter trigger input. The format of this input is expected to be Q0 with a value of either 0 or 7FFF.

Outputs: The output channel from this block component is the Modulo 6 counter output. The format of this output is Q0 in a range of 0-5.

Parameters:
The following parameters of 'Modulo6 Counter' may be modified:

Precision	This block component requires that the precision field be set for integer precision. This parameter is not user-selectable.

Note - All parameters except the 'Precision' parameter of Modulo6 Counter are able to be set by either the user, or by the output of any other DSP-based component.

Description:
This block component implements a modulo 6 counter. It counts from state 0 through 5, then resets to 0 and repeats the process. The state of the output variable m6_cntr changes to the next state every time it receives a trigger input through the input variable m6_trig_in.

The following equation is implemented:

m6_cntr= 0, when 1 st trigger pulse occur (m6_trig_in is set to 7FFF for the 1 st time)
= 1, when 2 nd trigger pulse occur (m6_trig_in is set to 7FFF for the 2 nd time)
= 2, when 3 rd trigger pulse occur (m6_trig_in is set to 7FFF for the 3 rd time)
= 3, when 4 th trigger pulse occur (m6_trig_in is set to 7FFF for the 4 th time)
= 4, when 5 th trigger pulse occur (m6_trig_in is set to 7FFF for the 5 th time)
= 5, when 6 th trigger pulse occur (m6_trig_in is set to 7FFF for the 6 th time)

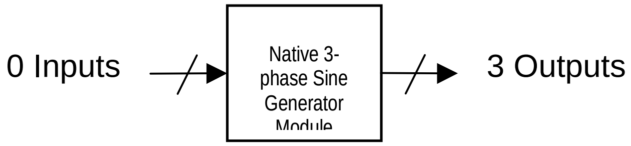
and repeats the output states for subsequent pulses.

Native 3-phase Sine Generator Module

DSP-based

RIDE	Block Diagram	ImageDSP	VAB for C2000	VAB for C5000	VAB for C6000	VAB for UNIV
●			●			

Component Name: Native 3-phase Sine Generator Module



Library	Digital Motor Control Library	
Group	Signal Generators	
File Name	RTPH3SIN.DLL	Revision: 4.00

Overview: Generates a 3-phase sine wave

Outputs: This block component will generate a three-phase sine wave on the three output channels. The phase is governed by the Step Angle user parameters.

Parameters:
The following parameters of 'Native 3-phase Sine Generator Module' may be modified:

Step Angle A The Step Angle A parameter is used to adjust the phase of the first sine wave output. The step angle is used as an offset into a sine lookup table.

Step Angle B The Step Angle B parameter is used to adjust the phase of the second sine wave output. The step angle is used as an offset into a sine lookup table.

Step Angle C The Step Angle C parameter is used to adjust the phase of the third sine wave output. The step angle is used as an offset into a sine lookup table.

Precision This block component requires that the precision field be set for integer precision. This parameter is not user-selectable.

Note - All parameters except the 'Precision' parameter of Native 3-phase Sine Generator Module are able to be set by either the user, or by the output of any other DSP-based component.

Description:

This block component will generates a 3-phase sine waveform that is based upon three Step Angle user parameters. These step angles are used as offsets into a sine lookup table to produce the 2-phase sine output.

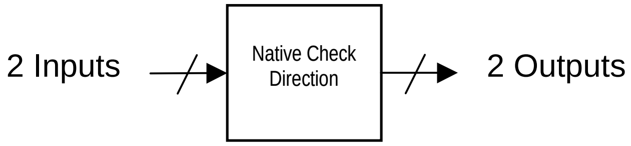
> **Note for 'Native 3-phase Sine Generator Module':**
> **Since this is a Native function it has been optimized for single sample processing and is quite efficient, especially for control applications and systems which require small amounts of processing delay.**
>
> **Although it will interface with other non-Native functions (i.e., Vector functions), the inputs and outputs of this component are treated as scalar values and not as vectors, so this function will only process the first sample of a vector fed into it, and its output framesize is always inherently 1.**

Native Check Direction

DSP-based

RIDE	Block Diagram	ImageDSP	VAB for C2000	VAB for C5000	VAB for C6000	VAB for UNIV
●			●			

Component Name: Native Check Direction



Library	Digital Motor Control Library	
Group	General Functions	
File Name	RTCHGDIR.DLL	Revision: 4.00

Overview: Applies sign-conversion to the input data based upon an input reference signal

Inputs: The first input channel to this block component is used as a reference signal. The second input channel is the actual data to undergo processing. If the reference signal is greater than or equal to zero then the block component will pass through the data on its second input channel with no change. If the reference signal is negative then the data on the block's second channel will be negated.

Outputs: The first output channel provides the sign bit of the process. The second output channel contains the processed data. If the reference signal was positive and no sign-conversion took place then this sign-bit output will be a 1. If the reference signal was negative then sign-conversion did take place and the sign-bit output will be a -1.

Parameters:
The following parameters of 'Native Check Direction' may be modified:

Precision	This block component requires that the precision field be set for integer precision. This parameter is not user-selectable.

Note - All parameters except the 'Precision' parameter of Native Check Direction are able to be set by either the user, or by the output of any other DSP-based component.

Description:
This block component allows for the sign-conversion of an input signal. The basis for negating the input signal is governed by an input reference signal. If the reference signal is positive then no sign change ocurrs, and the input signal is generated on the block's output channel. If the reference signal is negative then the input signal is negated and generated on the block's output channel. A sign bit is also generated for reference purposes.

Note for 'Native Check Direction':
Since this is a Native function it has been optimized for single sample processing and is quite efficient, especially for control applications and systems which require small amounts of processing delay.

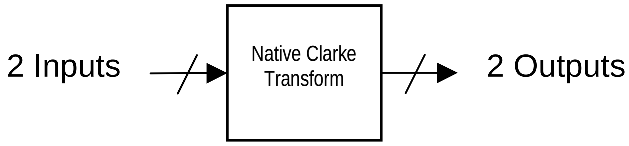
Although it will interface with other non-Native functions (i.e., Vector functions), the inputs and outputs of this component are treated as scalar values and not as vectors, so this function will only process the first sample of a vector fed into it, and its output framesize is always inherently 1.

Native Clarke Transform

DSP-based

RIDE	Block Diagram	ImageDSP	VAB for C2000	VAB for C5000	VAB for C6000	VAB for UNIV
●			●			

Component Name: Native Clarke Transform



Library	Digital Motor Control Library	
Group	Transforms	
File Name	NTCLARK.DLL	Revision: 4.00

Overview: Performs Clarke transform

Inputs: The first input channel to this block component is the Phase ‘a’ component of the balanced three phase quantities. The second input channel input channel is the Phase ‘b’ component of the balanced three phase quantities. The format for both inputs is expected to be Q15 with a range of 8000-7FFF.

Outputs: The first output channel to this block component is the Direct axis(d) component of the transformed signal. The second output channel is the Quadrature axis(q) component of the transformed signal. The format for both outputs is Q15 with a range of 8000-7FFF.

Parameters:
The following parameters of 'Native Clarke Transform' may be modified:

Precision	This block component requires that the precision field be set for integer precision. This parameter is not user-selectable.

Note - All parameters except the 'Precision' parameter of Native Clarke Transform are able to be set by either the user, or by the output of any other DSP-based component.

Description:
This block component uses the Clarke Transform to convert balanced three phase quantities into balanced two phase quadrature quantities. The input Phase components are used as an input to the Clarke Transform to calculate the Direct Axis and Quadrature Axis output components.

Note for 'Native Clarke Transform':
Since this is a Native function it has been optimized for single sample processing and is quite efficient, especially for control applications and systems which require small amounts of processing delay.

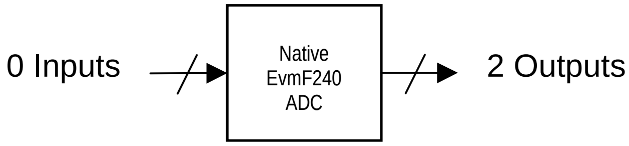
Although it will interface with other non-Native functions (i.e., Vector functions), the inputs and outputs of this component are treated as scalar values and not as vectors, so this function will only process the first sample of a vector fed into it, and its output framesize is always inherently 1.

Native EvmF240 ADC

DSP-based

RIDE	Block Diagram	ImageDSP	VAB for C2000	VAB for C5000	VAB for C6000	VAB for UNIV
●			●			

Component Name: Native EvmF240 ADC



Library	Digital Motor Control Library	
Group	EvmF240 Hardware Functions	
File Name	NTADC24.DLL	Revision: 4.00

Overview: Performs analog-to-digital conversion on EVM F240

Outputs: This block component's first output channel is the sampled data from ADC1. The actual ADC channel acquired is specified by the ADC1 Channel user-parameter. The second output channel is the sampled data from ADC2. The actual ADC channel acquired is specified by the ADC2 Channel user-parameter. The Q-scaling format for the output channels is specified by the Precision user-parameter.

Parameters:
The following parameters of 'Native EvmF240 ADC' may be modified:

ADC1 Channel The ADC1 Channel parameter is used to select the ADC channel that is to be sampled and used as the block component's output channel 1. ADC channel options include Channels 1-8.

ADC2 Channel The ADC2 Channel parameter is used to select the ADC channel that is to be sampled and used as the block component's output channel 2. ADC channel options include Channels 9-16.

Precision This block component requires that the precision field be chosen from a list of Q formats. The range of Q formats available are from Q0 to Q15. Selecing a Q format will result in the block component's output channels to also be in the same format.

Note - All parameters except the 'Precision' parameter of Native EvmF240 ADC are able to be set by either the user, or by the output of any other DSP-based component.

Description:
This block component provides for EVM F240 2-channel analog-to-digital conversion with selectable channel pairing.

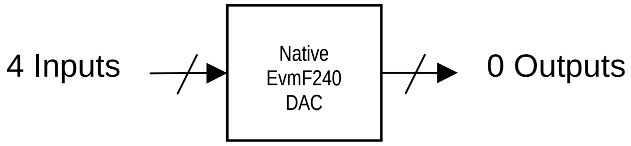
> **Note for 'Native EvmF240 ADC':**
> **Since this is a Native function it has been optimized for single sample processing and is quite efficient, especially for control applications and systems which require small amounts of processing delay.**
>
> **Although it will interface with other non-Native functions (i.e., Vector functions), the inputs and outputs of this component are treated as scalar values and not as vectors, so this function will only process the first sample of a vector fed into it, and its output framesize is always inherently 1.**

Native EvmF240 DAC

DSP-based

RIDE	Block Diagram	ImageDSP	VAB for C2000	VAB for C5000	VAB for C6000	VAB for UNIV
●			●			

Component Name: Native EvmF240 DAC



Library	Digital Motor Control Library	
Group	EvmF240 Hardware Functions	
File Name	NTDAC24.DLL	Revision: 4.00

Overview: Performs digital-to-analog conversion on EVM F240

Inputs: The four input channels to this block component represent the data to undergo D/A conversion. These input channels will be applied to the DAC of the EVM F240 DSP board.

Parameters:
The following parameters of 'Native EvmF240 DAC' may be modified:

Precision	This block component requires that the precision field be set for integer precision. This parameter is not user-selectable.

Note - All parameters except the 'Precision' parameter of Native EvmF240 DAC are able to be set by either the user, or by the output of any other DSP-based component.

Description:
This block component provides for EVM F240 4-channel digital-to-analog conversion. This block component will convert 4 input channels of data into its analog equivalent. The output channels are applied to the DAC on the EVM F240 DSP board.

Note for 'Native EvmF240 DAC':
Since this is a Native function it has been optimized for single sample processing and is quite efficient, especially for control applications and systems which require small amounts of processing delay.

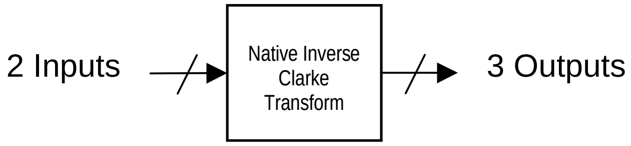
Although it will interface with other non-Native functions (i.e., Vector functions), the inputs and outputs of this component are treated as scalar values and not as vectors, so this function will only process the first sample of a vector fed into it, and its output framesize is always inherently 1.

Native Inverse Clarke Transform

DSP-based

RIDE	Block Diagram	ImageDSP	VAB for C2000	VAB for C5000	VAB for C6000	VAB for UNIV
●			●			

Component Name: Native Inverse Clarke Transform



Library	Digital Motor Control Library	
Group	Transforms	
File Name	NTICLARK.DLL	Revision: 4.00

Overview: Performs Inverse Clarke Transform

Inputs: The first input channel to this block component is the Direct axis(d) component of the input two phase signal. The second input channel is the Quadrature axis(q) component of the input two phase signal. Both inputs are expected to be Q15 format with a range of 8000-7FFF.

Outputs: The first output channel from this block component is the Phase ‘a’ component of the transformed signal. The second output channel is the Phase ‘b’ component of the transformed signal. The third output channel is the Phase ‘c’ component of the transformed signal. All outputs are in Q15 format with a range of 8000-7FFF.

Parameters:
The following parameters of 'Native Inverse Clarke Transform' may be modified:

Precision This block component requires that the precision field be set for integer precision. This parameter is not user-selectable.

Note - All parameters except the 'Precision' parameter of Native Inverse Clarke Transform are able to be set by either the user, or by the output of any other DSP-based component.

Description:
This block component uses an Inverse Clarke Transform to convert balanced two phase quadrature quantities into balanced three phase quantities. The input Direct axis and Quadrature axis componentsare used as an input to the inverse Clarke Transform to calculate the three phase output components.

Note for 'Native Inverse Clarke Transform':

Since this is a Native function it has been optimized for single sample processing and is quite efficient, especially for control applications and systems which require small amounts of processing delay.

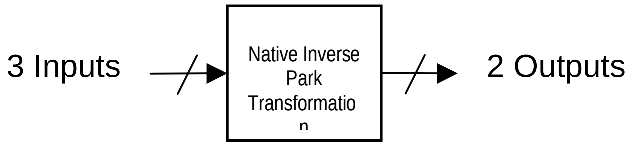
Although it will interface with other non-Native functions (i.e., Vector functions), the inputs and outputs of this component are treated as scalar values and not as vectors, so this function will only process the first sample of a vector fed into it, and its output framesize is always inherently 1.

Native Inverse Park Transformation

DSP-based

RIDE	Block Diagram	ImageDSP	VAB for C2000	VAB for C5000	VAB for C6000	VAB for UNIV
●			●			

Component Name: Native Inverse Park Transformation



Library	Digital Motor Control Library	
Group	Transforms	
File Name	NTIPARK.DLL	Revision: 4.00

Overview: Performs an Inverse Park Transformation

Inputs: The first input channel to this block component is the Direct axis(D) component of the input in a rotating reference frame. The second input channel is the Quadrature axis(Q) component of the input in a rotating reference frame. The third input channel is the Phase angle between the stationary and rotating frame. All inputs are expected to be in Q15 format with a range of 8000-7FFF.

Outputs: The first output channel from this block component is the Direct axis(d) component of the transformed signal in a stationary reference frame. The second output channel is the Quadrature axis(q) component of the transformed signal in a stationary reference frame. Both outputs are in Q15 format with a range of 8000-7FFF.

Parameters:
The following parameters of 'Native Inverse Park Transformation' may be modified:

Precision	This block component requires that the precision field be set for integer precision. This parameter is not user-selectable.

Note - All parameters except the 'Precision' parameter of Native Inverse Park Transformation are able to be set by either the user, or by the output of any other DSP-based component.

Description:
This block component uses an Inverse Park Transformation to project vectors in an orthogonal rotating reference frame into a two phase orthogonal stationary frame.

Note for 'Native Inverse Park Transformation':

Since this is a Native function it has been optimized for single sample processing and is quite efficient, especially for control applications and systems which require small amounts of processing delay.

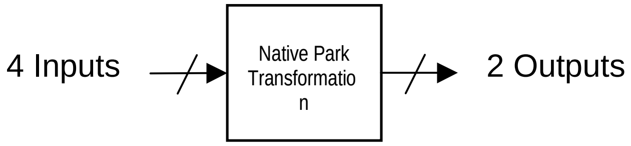
Although it will interface with other non-Native functions (i.e., Vector functions), the inputs and outputs of this component are treated as scalar values and not as vectors, so this function will only process the first sample of a vector fed into it, and its output framesize is always inherently 1.

Native Park Transformation

DSP-based

RIDE	Block Diagram	ImageDSP	VAB for C2000	VAB for C5000	VAB for C6000	VAB for UNIV
●			●			

Component Name: Native Park Transformation



Library	Digital Motor Control Library	
Group	Transforms	
File Name	NTPARK.DLL	Revision: 4.00

Overview: Performs Park Transformation

Inputs: The first input channel of this block component is the Direct axis(d) component of the input signal in stationary reference frame. The second input channel is the Quadrature axis(q) component of the input signal in stationary reference frame. The third input channel is the Phase angle between stationary and rotating frame. The format for these inputs is expected to be Q15 in the range of 8000-7FFF.

Outputs: The first output channel of this block component is the Direct axis(D) component of transformed signal in rotating reference frame. The second channel is the Quadrature axis(Q) component of transformed signal in rotating reference frame. Both outputs are in Q15 format with a range of 8000-7FFF.

Parameters:
The following parameters of 'Native Park Transformation' may be modified:

Precision This block component requires that the precision field be set for integer precision. This parameter is not user-selectable.

Note - All parameters except the 'Precision' parameter of Native Park Transformation are able to be set by either the user, or by the output of any other DSP-based component.

Description:
This block component uses the Park Transformation to convert vectors in a balanced 2-phase orthogonal stationary system into an orthogonal rotating reference frame. The Direct & Quadrature stationary reference frame axis, and a Phase angle are used as an input to the Park Transform to calculate the Direct & Quadrature rotating reference frame axis output components.

Note for 'Native Park Transformation':

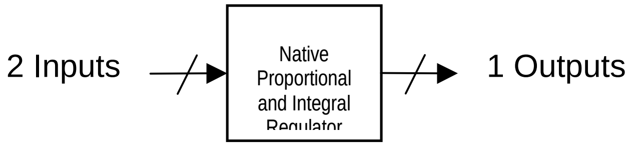
Since this is a Native function it has been optimized for single sample processing and is quite efficient, especially for control applications and systems which require small amounts of processing delay.

Although it will interface with other non-Native functions (i.e., Vector functions), the inputs and outputs of this component are treated as scalar values and not as vectors, so this function will only process the first sample of a vector fed into it, and its output framesize is always inherently 1.

Native Proportional and Integral Regulator

						DSP-based
RIDE	Block Diagram	ImageDSP	VAB for C2000	VAB for C5000	VAB for C6000	VAB for UNIV
●			●			

Component Name: Native Proportional and Integral Regulator



Library	Digital Motor Control Library	
Group	Arithmetic Functions	
File Name	NTPI.DLL	Revision: 4.00

Overview: Implements a Proportional and Integral regulator

Inputs: The first input to this PI block component is the input signal. The second input is the reference signal. The format for both inputs is expected to be Q15 in the range of 8000-7FFF.

Outputs: The output of this PI regulator block component is a Q15 number in the range of 8000-7FFF.

Parameters:
The following parameters of 'Native Proportional and Integral Regulator' may be modified:

Proportional Term (Kp) The Proportional Term (Kp) parameter allows for selection of the Proportional gain coefficient. The format of this parameter is expected to be Q15.

Integral Term (Ki) The Integral Term (Ki) parameter is used to specify the Integral coefficient. The format is expected to be Q15 with the range being system dependent.

Saturation Term (Ksat) The Saturation Term (Ks) parameter is used to specify the saturation coefficient. The format is expected to be Q15 with the range being system dependent.

Precision This block component requires that the precision field be set for integer precision. This parameter is not user-selectable.

Note - All parameters except the 'Precision' parameter of Native Proportional and Integral Regulator are able to be set by either the user, or by the output of any other DSP-based component.

Description:
This block component implements a classical numerical PI regulator.

Note for 'Native Proportional and Integral Regulator':
Since this is a Native function it has been optimized for single sample processing and is quite efficient, especially for control applications and systems which require small amounts of processing delay.

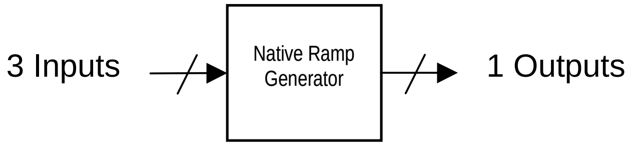
Although it will interface with other non-Native functions (i.e., Vector functions), the inputs and outputs of this component are treated as scalar values and not as vectors, so this function will only process the first sample of a vector fed into it, and its output framesize is always inherently 1.

Native Ramp Generator

DSP-based

RIDE	Block Diagram	ImageDSP	VAB for C2000	VAB for C5000	VAB for C6000	VAB for UNIV
●			●			

Component Name: Native Ramp Generator



Library	Digital Motor Control Library
Group	Signal Generators
File Name	NTRAMP.DLL Revision: 4.00

Overview: Generates an output ramp based upon input data values

Inputs: The three inputs to this block component are used to govern the generated ramp output signal. The first channel is the Gain value. The second channel is the desired Frequency value. The third channel is the Offset value. The format for these input channels is expected to be Q15.

Outputs: The output channel of this block component is a Normalized Ramp. The format of the output is Q15.

Parameters:
The following parameters of 'Native Ramp Generator' may be modified:

Precision	This block component requires that the precision field be set for integer precision. This parameter is not user-selectable.

Note - All parameters except the 'Precision' parameter of Native Ramp Generator are able to be set by either the user, or by the output of any other DSP-based component.

Description:
This block component will generate a Ramp output. The parameters used in calculating this output ramp signal is governed by the block's three input data channels. The Gain, Frequency, and Offset parameters are represented by these input channels.

Note for 'Native Ramp Generator':
Since this is a Native function it has been optimized for single sample processing and is quite efficient, especially for control applications and systems which require small amounts of processing delay.

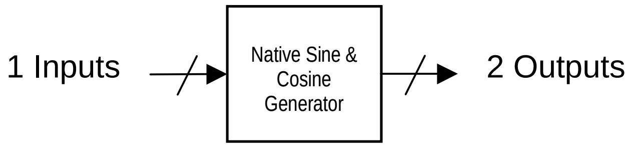
Although it will interface with other non-Native functions (i.e., Vector functions), the output of this component is a scalar value and not as a vector, so its output framesize is always inherently 1.

Native Sine & Cosine Generator

DSP-based

RIDE	Block Diagram	ImageDSP	VAB for C2000	VAB for C5000	VAB for C6000	VAB for UNIV
•			•			

Component Name: Native Sine & Cosine Generator



Library	Digital Motor Control Library	
Group	Signal Generators	
File Name	NTSINCOS.DLL	Revision: 4.00

Overview: Sine and Cosine waveform generation

Inputs: The block component's input channel is used as a data offset into a sine lookup table. Varying this input will allow for frequency modification of the sine and cosine waveform.

Outputs: The block component's first output channel is the generated sine waveform data. The second output channel is the generated cosine waveform data. The data format is Q15.

Parameters:
The following parameters of 'Native Sine & Cosine Generator' may be modified:

Precision	This block component requires that the precision field be set for integer precision. This parameter is not user-selectable.

Note - All parameters except the 'Precision' parameter of Native Sine & Cosine Generator are able to be set by either the user, or by the output of any other DSP-based component.

Description:
This block component will generate both a sine and cosine waveform that is based upon the data block component's input data channel. The resulting Q15 sine and cosine waveforms are computed by using the input data as an offset into a sine lookup table. To obtain both sine and cosine from the angle teta, a sine lookup table has been implemented. The table contains 256 words to represent sine values of electrical angles in the range [0 ; 360]. As a result, the resolution on q is limited to 360/256 =1.40625 .

q = Angle / 360 (with q in the range [0;FFFh]) q varies from 0 to 3276. As only 256 words are available to represent this range, q is divided by 256 and stored into the variable index that will be used to address the lookup table.

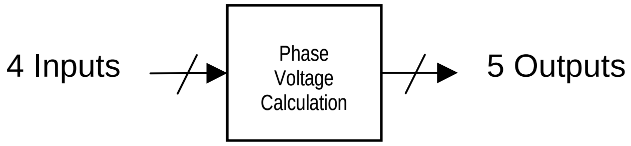
> **Note for 'Native Sine & Cosine Generator':**
> **Since this is a Native function it has been optimized for single sample processing and is quite efficient, especially for control applications and systems which require small amounts of processing delay.**
>
> **Although it will interface with other non-Native functions (i.e., Vector functions), the inputs and outputs of this component are treated as scalar values and not as vectors, so this function will only process the first sample of a vector fed into it, and its output framesize is always inherently 1.**

Phase Voltage Calculation

DSP-based

RIDE	Block Diagram	ImageDSP	VAB for C2000	VAB for C5000	VAB for C6000	VAB for UNIV
●			●			

Component Name: Phase Voltage Calculation



Library	Digital Motor Control Library	
Group	AC Induction Motor Functions	
File Name	PHASEVC.DLL	Revision: 4.00

Overview: Phase Voltage Calculation

Inputs: The first input channel to this block component is the DC-bus voltage. The last three input channels are the switching functions of the upper switching devices (1,2, & 3). All input values are expected to be Q15 data format.

Outputs: The first three output channels from this block component are the Line-neutral phase voltages (A, B, & C). The fourth output channel is the stationary d-axis phase voltage. The fifth output channel is the stationary q-axis phase voltage. All outputs are Q15 data format.

Parameters:
The following parameters of 'Phase Voltage Calculation' may be modified:

Precision This block component requires that the precision field be set for integer precision. This parameter is not user-selectable.

Note - All parameters except the 'Precision' parameter of Phase Voltage Calculation are able to be set by either the user, or by the output of any other DSP-based component.

Description:
This block component calculates three phase voltages applied to the 3-ph motor (i.e., induction or synchronous motor) using the conventional voltage-source inverter. Three phase voltages can be reconstructed from the DC-bus voltage and three switching functions of the upper power switching devices of the inverter. In addition, this block component also includes the Clarke Transformation that converts three phase voltages into two stationary dq-axis voltages.

The phase voltage of a general 3-ph motor (V an , V bn , and V cn) can be calculated from the DC- us voltage (V dc) and three upper switching functions of inverter (S 1 , S 2 , and S

3). The 3-ph windings of motor are connected either D or U without a neutral return path (or 3-ph, 3-wire system).

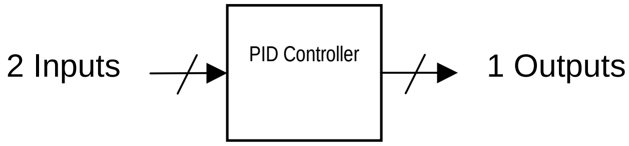
Each phase of the motor is simply modeled as a series impedance of resistance and inductance (r, L) and back emf (e a , e b , e c).

PID Controller

DSP-based

RIDE	Block Diagram	ImageDSP	VAB for C2000	VAB for C5000	VAB for C6000	VAB for UNIV
●			●			

Component Name: PID Controller



Library	Digital Motor Control Library	
Group	AC Induction Motor Functions	
File Name	PID_REG.DLL	Revision: 4.00

Overview: Implements a PID Control Function for F24xx

Inputs: The first input channel to this block component is the Reference signal. The second input channel is the Error signal. Both inputs are expected to be in Q15 format.

Outputs: The PID output channel from this block component is computed to be in Q15 format.

Parameters:
The following parameters of 'PID Controller' may be modified:

K0	This parameter specifies the K0 PID Coeffiecient. The format for this parameter is expected to be Q15.
K1	This parameter specifies the K1 PID Coeffiecient. The format for this parameter is expected to be Q15.
K2	This parameter specifies the K2 PID Coeffiecient. The format for this parameter is expected to be Q15.
Kx	This parameter specifies the Kx PID Coeffiecient. The format for this parameter is expected to be Q15.
Kf	This parameter specifies the Kf PID Coeffiecient. The format for this parameter is expected to be Q15.
psat	This parameter specifies the psat PID Coeffiecient. The format for this parameter is expected to be Q15.

nsat This parameter specifies the nsat PID Coeffiecient. The format for this parameter is expected to be Q15.

Precision This block component requires that the precision field be set for integer precision. This parameter is not user-selectable.

Note - All parameters except the 'Precision' parameter of PID Controller are able to be set by either the user, or by the output of any other DSP-based component.

Description:
Implements a PID Control Function for F24xx

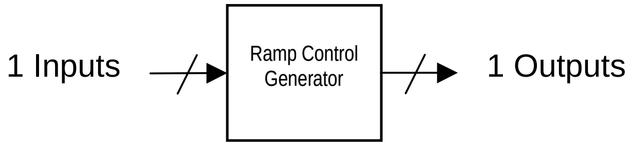
Note for 'PID Controller':
Since this is a Native function it has been optimized for single sample processing and is quite efficient, especially for control applications and systems which require small amounts of processing delay.

Although it will interface with other non-Native functions (i.e., Vector functions), the inputs and outputs of this component are treated as scalar values and not as vectors, so this function will only process the first sample of a vector fed into it, and its output framesize is always inherently 1.

Ramp Control Generator

DSP-based						
RIDE	Block Diagram	ImageDSP	VAB for C2000	VAB for C5000	VAB for C6000	VAB for UNIV
•			•			

Component Name: Ramp Control Generator



Library	Digital Motor Control Library
Group	Signal Generators
File Name	RAMP_CNTL.DLL Revision: 4

Overview: Generates a ramp-up and ramp-down function

Inputs: The input channel to this block component is the desired value of the ramp. The data format for this input is expected to be Q0 and is bounded by the Ramp Low Limit & Ramp High Limit user-parameters.

Outputs: The first output channel from this block component is the Ramp output value. The data format for this output is Q0 and lies within a range between the Ramp Low Limit & Ramp High Limit user-parameters. The second output channel is the Ramp output status flag. The format for this output channel is Q0 and will be either a 0 or 7FFF value.

Parameters:
The following parameters of 'Ramp Control Generator' may be modified:

Ramp Delay Max	The Ramp Delay Max paramater is used to specify the Ramp step delay in number of sampling cycles. The format for this parameter is expected to be Q0 with a range of 0-7FFF.
Ramp Low Limit	The Ramp Low Limit parameter is used to specify the Minimum value of ramp. The data format for this parameter is expected to be Q0 and within the range of 0-7FFF.
Ramp High Limit	The Ramp High Limit parameter is used to specify the Maximum value of ramp. The data format for this parameter is expected to be Q0 and within the range of 0-7FFF.

Precision	This block component requires that the precision field be set for integer precision. This parameter is not user-selectable.

Note - All parameters except the 'Precision' parameter of Ramp Control Generator are able to be set by either the user, or by the output of any other DSP-based component.

Description:

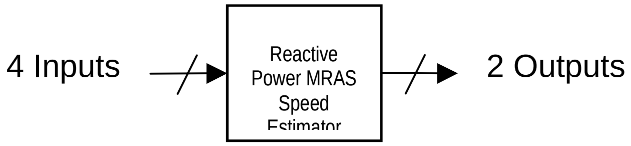
This block component implements a ramp-up and ramp-down function. The second output channel data (s_eq_t_flg) is set to 7FFF whenever the first output channel data (setpt_value) is equal to the input channel (target_value).

Reactive Power MRAS Speed Estimator

DSP-based

RIDE	Block Diagram	ImageDSP	VAB for C2000	VAB for C5000	VAB for C6000	VAB for UNIV
●			●			

Component Name: Reactive Power MRAS Speed Estimator



Library	Digital Motor Control Library	
Group	AC Induction Motor Functions	
File Name	ACI_MRAS.DLL	Revision: 4

Overview: Implements a speed estimator for the 3-ph induction motor

Inputs: The first input channel to this block component is the Stationary alfa-axis stator voltage (pu). The second input channel is the Stationary beta-axis stator voltage (pu). The third input channel is the Stationary alfa-axis stator current (pu). The fourth input channel is the Stationary beta-axis stator current (pu). The data format for these inputs is expected to be Q15.

Outputs: The first output channel from this block component is the Estimated rotor speed (pu). The data format for this output is Q15. The second output channel is the Estimated rotor speed (rpm). The data format for this output is Q0.

Parameters:
The following parameters of 'Reactive Power MRAS Speed Estimator' may be modified:

K1 The K1 parameter is computed using the machine parameters (Ls, Lr, Lm, Tr), base quantities (Ib, Vb, Wb), and sampling period (T). The data format for this parameter is Q10 (-32 to 31.999).

K1 = (Ls–Lm^2/Lr)*Ib/(T*Vb)

K2 The K2 parameter is computed using the machine parameters (Ls, Lr, Lm, Tr), base quantities (Ib, Vb, Wb), and sampling period (T). The data format for this parameter is Q15 (-1 to 0.999).

K2 = Lm^2*Ib/(Lr*Tr*Vb)

K3

The K3 parameter is computed using the machine parameters (Ls, Lr, Lm, Tr), base quantities (Ib, Vb, Wb), and sampling period (T). The data format for this parameter is Q8 (-128 to 127.996).

K3 = Tr*Wb

K4

The K4 parameter is computed using the machine parameters (Ls, Lr, Lm, Tr), base quantities (Ib, Vb, Wb), and sampling period (T). The data format for this parameter is Q15 (-1 to 0.999).

K4 = (Wb*T)^2/2

K5

The K5 parameter is computed using the machine parameters (Ls, Lr, Lm, Tr), base quantities (Ib, Vb, Wb), and sampling period (T). The data format for this parameter is Q15 (-1 to 0.999).

K5 = 1–T/Tr+T^2/(2*Tr^2)

K6

The K6 parameter is computed using the machine parameters (Ls, Lr, Lm, Tr), base quantities (Ib, Vb, Wb), and sampling period (T). The data format for this parameter is Q15 (-1 to 0.999).

K6 = Wb*(T–T^2/Tr)

K7

The K7 parameter is computed using the machine parameters (Ls, Lr, Lm, Tr), base quantities (Ib, Vb, Wb), and sampling period (T). The data format for this parameter is Q15 (-1 to 0.999).

K7 = T/Tr–T^2/(2*Tr^2)

Base RPM

The Base RPM parameter is used to specify the base RPM value. The data format for this parameter is Q3 (-4096 to 4095.9).

base_rpm = 120*base_freq/no_poles

Precision

This block component requires that the precision field be set for integer precision. This parameter is not user-selectable.

Note - All parameters except the 'Precision' parameter of Reactive Power MRAS Speed Estimator are able to be set by either the user, or by the output of any other DSP-based component.

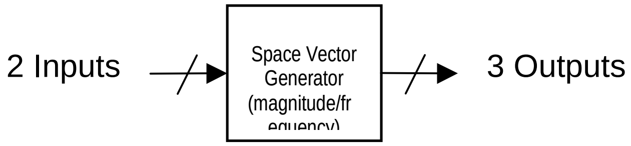
Description:
This block component implements a speed estimator for the 3-ph induction motor based on reactive power model reference adaptive system (MRAS). In this technique, there are two subsystems called reference and adaptive models, which compute the reactive power of the induction motor. Since both pure integrators and stator resistance are not associated in the reference model, the reactive power MRAS is independent of initial conditions and insensitive to variation of stator resistance.

Space Vector Generator (magnitude/frequency)

DSP-based

RIDE	Block Diagram	ImageDSP	VAB for C2000	VAB for C5000	VAB for C6000	VAB for UNIV
●			●			

Component Name: Space Vector Generator (magnitude/frequency)



Library	Digital Motor Control Library	
Group	AC Induction Motor Functions	
File Name	SVGEN_MF.DLL	Revision: 4.00

Overview: Calculates the appropriate duty ratios needed to generate a given stator reference voltage using space vector PWM technique.

Inputs: The first input represents the fraction of frequency of reference voltage vector in a Q15 format (8000-7FFF). The second input channel represents the required gain for the desired reference voltage vector in a Q15 format (8000-7FFF)

Outputs: The three output channels represent Phase A output Ta, Phase B output Tb, & Phase C output Tc. This corresponds to the Duty ratio of PWM1, the Duty ratio of PWM3, and the Duty ratio of PWM5. These output channels are represented in Q15 format (8000-7FFF)

Parameters:
The following parameters of 'Space Vector Generator (magnitude/frequency)' may be modified:

Max Frequency The Max Frequency parameter allows for selection of the reference voltage vector freqeuncy. This value is entered as a Q0 number (8000-7FFF)

Precision This block component requires that the precision field be set for integer precision. This parameter is not user-selectable.

Note - All parameters except the 'Precision' parameter of Space Vector Generator (magnitude/frequency) are able to be set by either the user, or by the output of any other DSP-based component.

Description:
This block component calculates the appropriate duty ratios needed to generate a given stator reference voltage using space vector PWM technique. The stator reference voltage is described by it's (a,b) components, Ualfa and Ubeta.

The Space Vector Pulse Width Modulation (SVPWM) refers to a special switching sequence of the upper three power devices of a three-phase voltage source inverters (VSI) used in application such as AC induction and permanent magnet synchronous motor drives. This special switching scheme for the power devices results in 3 pseudo-sinusoidal currents in the stator phases.

It has been shown that SVPWM generates less harmonic distortion in the output voltages or currents in the windings of the motor load and provides more efficient use of DC supply voltage, in comparison to direct sinusoidal modulation technique.

Note for 'Space Vector Generator (magnitude/frequency)':
Since this is a Native function it has been optimized for single sample processing and is quite efficient, especially for control applications and systems which require small amounts of processing delay.

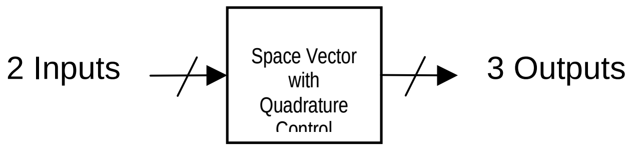
Although it will interface with other non-Native functions (i.e., Vector functions), the inputs and outputs of this component are treated as scalar values and not as vectors, so this function will only process the first sample of a vector fed into it, and its output framesize is always inherently 1.

Space Vector with Quadrature Control

DSP-based

RIDE	Block Diagram	ImageDSP	VAB for C2000	VAB for C5000	VAB for C6000	VAB for UNIV
●			●			

Component Name: Space Vector with Quadrature Control



Library	Digital Motor Control Library	
Group	AC Induction Motor Functions	
File Name	SVGEN_DQ.DLL	Revision: 4.00

Overview: Calculate the duty ratios needed to generate a given stator reference voltage

Inputs: The first input channel to this block component is the component of reference stator voltage vector on the direct axis stationary reference frame. The second input channel is the component of reference stator voltage vector on the quadrature axis stationary reference frame. Both inputs are expected to be in Q15 format within a range of 8000-7FFF.

Outputs: The first output channel from this block component is the duty ratio of PWM1 (CMPR1 register value as a fraction of associated period register, TxPR, value). The second output channel is the duty ratio of PWM3 (CMPR2 register value as a fraction of associated period register, TxPR, value). The third output channel is the duty ratio of PWM5 (CMPR3 register value as a fraction of associated period register, TxPR, value). All outputs are in Q15 format within a range of 8000-7FFF.

Parameters:
The following parameters of 'Space Vector with Quadrature Control' may be modified:

Precision This block component requires that the precision field be set for integer precision. This parameter is not user-selectable.

Note - All parameters except the 'Precision' parameter of Space Vector with Quadrature Control are able to be set by either the user, or by the output of any other DSP-based component.

Description:
This block component calculates the appropriate duty ratios needed to generate a given stator reference voltage using space vector PWM technique. The stator reference voltage is described by it's (a,b) components, Ualfa and Ubeta.

The Space Vector Pulse Width Modulation (SVPWM) refers to a special switching sequence of the upper three power devices of a three-phase voltage source inverters (VSI) used in application such as AC induction and permanent magnet synchronous motor drives. This special switching scheme for the power devices results in 3 pseudo-sinusoidal currents in the stator phases.

It has been shown that SVPWM generates less harmonic distortion in the output voltages or currents in the windings of the motor load and provides more efficient use of DC supply voltage, in comparison to direct sinusoidal modulation technique.

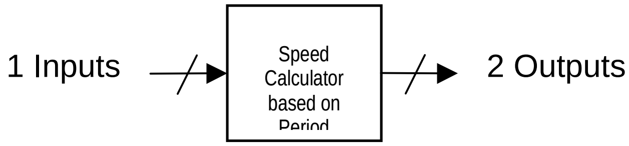
Note for 'Space Vector with Quadrature Control':
The framesize for both input channels is expected to be of length 1. The output framesize is also of length 1.

Speed Calculator based on Period Measurement

DSP-based

RIDE	Block Diagram	ImageDSP	VAB for C2000	VAB for C5000	VAB for C6000	VAB for UNIV
●			●			

Component Name: Speed Calculator based on Period Measurement



Library	Digital Motor Control Library	
Group	AC Induction Motor Functions	
File Name	SPEED_PRD.DLL	Revision: 4.00

Overview: Calculates the motor speed based on a signal's period measurement.

Inputs: The input channel of this block component is the captured base timer counter value corresponding to the periodic edges of the sensed signal. The format of the input is expected to be Q0 with a range of 0-FFFF.

Outputs: The first output channel represents the normalized motor speed. The format of this output is Q15 with a range of 0-7FFF. The second output channel represents the motor speed in revolution per minute. The format of this output is Q0 with a range of 0 to the Maximum RPM user-parameter.

Parameters:
The following parameters of 'Speed Calculator based on Period Measurement' may be modified:

PU Scaler	The PU Scaler parameter is used to convert 1/N cycles to a Q15-based speed.
Maximum RPM	The Maximum RPM parameter defines the speed of normalization. The value chosen for this parameter should be equal to or greater than the maximum motor speed. The format for this parameter is expected to be Q0.
Precision	This block component requires that the precision field be set for integer precision. This parameter is not user-selectable.

Note - All parameters except the 'Precision' parameter of Speed Calculator based on Period Measurement are able to be set by either the user, or by the output of any other DSP-based component.

Description:
This block component calculates the motor speed based on a signal's period measurement. Such a signal, for which the period is measured, can be the periodic output pulses from a motor speed sensor.

A low cost shaft sprocket with n teeth and a Hall effect gear tooth sensor is used to measure the motor speed. Fig. 1 shows the physical details associated with the sprocket. The Hall effect sensor outputs a square wave pulse every time a tooth rotates within its proximity. The resultant pulse rate is n pulses per revolution. The Hall effect sensor output is fed directly to the 24x/24xx Capture input pin. The capture unit will capture (the value of it's base timer counter) on either the rising or the falling edges(whichever is specified) of the Hall effect sensor output. The captured value is passed to this s/w module through the variable called time_stamp.

In this module, every time a new input time_stamp becomes available it is compared with the previous time_stamp. Thus, the tooth-to-tooth period (t2-t1) value is calculated. In order to reduce jitter or period fluctuation, an average of the most recent n period measurements can be performed each time a new pulse is detected.

Fig. 1 Speed measurement with a sprocket

From the two consecutive time_stamp values the difference between the captured values are calculated as,

? = time_stamp(new) – time_stamp(old)

Then the time period in sec is given by,

where,

KP = Prescaler value for the Capture unit time base
T = CPU clock period in sec
From Fig. 1, the angle ? in radian is given by,
where,
n = number of teeth in the sprocket
Then the speed ? in radian/sec and the normalized speed ?N are calculated as,

For, n=25, KP=32 and T=50x10-9 sec (20MHz CPU clock), the normalized speed ?N is given by,

The speed N in rpm is calculated as,

The capture unit in 24x/24xx allows accurate time measurement (in multiples of clock cycles and defined by a prescaler selection) between events. In this case the events are selected to be the rising edge of the incoming pulse train. What we are interested in is the delta time between events and hence for this implementation Timer 1 is allowed to free run with a prescale of 32 (1.6uS resolution) and the delta time ?, in scaled clock counts, is calculated as shown in Fig. 2.

Fig. 2 Calculation of speed

In Fig. 2, the vertical axis f(t) represents the value of the Timer counter which is running in continuous up count mode and resetting when the period register = 7FFFh. This allows ? to be calculated in Q15 format for later use in the averaging and division algorithms. Note that two cases need to be accounted for: the simple case where the Timer has not wrapped around and where it has wrapped around. By keeping the current and previous capture values it is easy to test for each of these cases.
Once a “robust” period measurement is extracted from the averaging algorithm, the speed is calculated using the appropriate equations explained before. In order to maintain high precision in the calculation for the full range of motor speeds, a 32-bit/16bit division is performed as shown in Fig. 3 in the following.

Fig. 3 32-bit/16bit division
Once complete the result is a 32-bit value in Q16 format containing both an integer and fractional part. This value is subsequently scaled to a 16 bit, Q15 format value for later calculation of the speed error (see Fig. 3).

Variable Cross Ref Table

Variables in the equations	Variables in the code
?	event_period
?N	speed_prd
N	speed_rpm

Note for 'Speed Calculator based on Period Measurement':
Since this is a Native function it has been optimized for single sample processing and is quite efficient, especially for control applications and systems which require small amounts of processing delay.

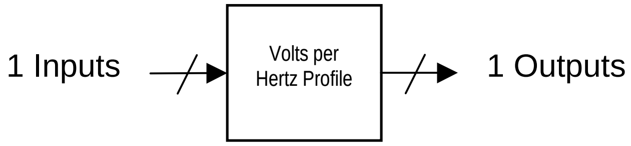
Although it will interface with other non-Native functions (i.e., Vector functions), the inputs and outputs of this component are treated as scalar values and not as vectors, so this function will only process the first sample of a vector fed into it, and its output framesize is always inherently 1.

Volts per Hertz Profile

DSP-based

RIDE	Block Diagram	ImageDSP	VAB for C2000	VAB for C5000	VAB for C6000	VAB for UNIV
•			•			

Component Name: Volts per Hertz Profile



Library	Digital Motor Control Library	
Group	AC Induction Motor Functions	
File Name	V_HZ_PROFILE.DLL	Revision: 4.00

Overview: Generates the output command voltage for a specific input command frequency according to specified V/Hz profile.

Inputs: The input to the Volts per Hertz Profile block component represents the Command frequency of the stator and is expected to be in Q15 format (0-7FFF)

Outputs: The output of the Volts per Hertz Profile block component represents the computed output voltage. This output voltage is dependent upon the Vmax, Vmin, FH, FL, and slope user parameters and is represented in Q15 format (0-7FFF)

Parameters:
The following parameters of 'Volts per Hertz Profile' may be modified:

Vmax The Vmax parameter allows you to specify the voltage that corresponds to the FH frequency. This voltage is the computed output for frequencies above FH. The Vmax parameter is entered as a Q15 number.

Vmin The Vmin parameter allows you to specify the voltage that corresponds to the FL frequency. This voltage is the computed output for frequencies below FL. The Vmin parameter is entered as a Q15 number.

FH The FH parameter allows you to specify the high frequency point on v/f profile. This defines the point at which frequencies above FH will cause the output to be computed as the Vmax voltage value. The FH parameter should be entered as a Q15 number.

FL The FL parameter allows you to specify the low frequency point on v/f profile. This defines the point at which frequencies below FL will cause the output to be computed as the Vmin voltage value. The FL parameter should be entered as a Q15 number.

Slope The Slope parameter allows you to specify the slope of the v/f profile. The slope parameter should be entered as a Q12 number.

Precision This block component requires that the precision field be set for integer precision. This parameter is not user-selectable.

Note - All parameters except the 'Precision' parameter of Volts per Hertz Profile are able to be set by either the user, or by the output of any other DSP-based component.

Description:
This module generates the output command voltage for a specific input command frequency according to specified volts/hertz profile. This is used for variable speed implementation of AC induction motor drives.

Note for 'Volts per Hertz Profile':
Since this is a Native function it has been optimized for single sample processing and is quite efficient, especially for control applications and systems which require small amounts of processing delay.

Although it will interface with other non-Native functions (i.e., Vector functions), the inputs and outputs of this component are treated as scalar values and not as vectors, so this function will only process the first sample of a vector fed into it, and its output framesize is always inherently 1.